REPORT DOCUMENTATION PAGE

Form Approved

OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 20-03-2015	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 01-04-2013 – 31-03-2015

4. TITLE AND SUBTITLE

Study of Discussion Record Analysis Using Temporal Data Crystallization and Its Application to TV Scene Analysis

5a. CONTRACT NUMBER
FA2386-13-1-4044

5b. GRANT NUMBER
Grant AOARD-134044

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)

Katsumi Nitta

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Tokyo Institute of Technology
J2-53, 4259 Nagatsuta-cho, Midori-ku, Yokohama, Japan 226-8502

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/AFOSR/IOA(AOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AOARD-134044

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Methods to analyze discussion records have been explored and a basic framework developed. Discussion records were analyzed by two methods – Scene analysis and Logical analysis. Scene analysis focuses on topic change in the record and divides a record into several scenes using Temporal Data Crystallization (TDC). Then, by measuring the polarity of words and the volume of utterance, the atmosphere and emotion in the scene were analyzed. Logical analysis focuses on the role of phrases in utterances, and attaches logical labels to phrases based on Toulmin diagram. By these labels, logical structure of arguments in the discussion was extracted, its semantics were calculated based on Theory of Computational Argumentation and various argumentation skills of the discussion record was evaluated. In addition to discussion analysis, the scene analysis techniques were applied to caption data of TV programs to recognize change of scenes, which requires real time analysis.

15. SUBJECT TERMS

Argumentation, Discussion analysis, Scene analysis, Temporal data crystalization, Computational argumentation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Hiroshi Motoda, Ph. D.
U	U	U	SAR	27	**19b. TELEPHONE NUMBER** *(Include area code)* +81-42-511-2011

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **20 MAR 2015**	2. REPORT TYPE **Final**	3. DATES COVERED **01-04-2013 to 31-03-2015**

4. TITLE AND SUBTITLE **Study of Discussion Record Analysis Using Temporal Data Crystallization and Its Application to TV Scene Analysis**	5a. CONTRACT NUMBER **FA2386-13-1-4044**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER **61102F**
6. AUTHOR(S) **Katsumi Nitta**	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Tokyo Institute of Technology,J2-53,4259 Nagatsuta-cho,Midori-ku, Yokohama, Japan,JP,226-8502**	8. PERFORMING ORGANIZATION REPORT NUMBER **N/A**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **AOARD, UNIT 45002, APO, AP, 96338-5002**	10. SPONSOR/MONITOR’S ACRONYM(S) **AFRL/AFOSR/IOA(AOARD)**
	11. SPONSOR/MONITOR’S REPORT NUMBER(S) **AOARD-134044**

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Methods to analyze discussion records have been explored and a basic framework developed. Discussion records were analyzed by two methods ??? Scene analysis and Logical analysis. Scene analysis focuses on topic change in the record and divides a record into several scenes using Temporal Data Crystallization (TDC). Then, by measuring the polarity of words and the volume of utterance, the atmosphere and emotion in the scene were analyzed. Logical analysis focuses on the role of phrases in utterances, and attaches logical labels to phrases based on Toulmin diagram. By these labels, logical structure of arguments in the discussion was extracted, its semantics were calculated based on Theory of Computational Argumentation and various argumentation skills of the discussion record was evaluated. In addition to discussion analysis, the scene analysis techniques were applied to caption data of TV programs to recognize change of scenes, which requires real time analysis.

15. SUBJECT TERMS

Argumentation, Discussion analysis, Scene analysis, Temporal data crystalization, Computational argumentation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**27**	

Final Report for AOARD Grant FA2386-13-1-4044

Study of Discussion Record Analysis Using Temporal Data Crystallization and Its Application to TV Scene Analysis

March 31 2015

Name of Principal Investigators (PI and Co-PIs):

- e-mail address : nitta@dis.titech.ac.jp
- Institution : Tokyo Institute of Technology
- Mailing Address : J2-53, 4259 Nagatsuta-cho, Midori-ku, Yokohama, Japan 226-8502
- Phone : +81-45-924-5214
- Fax : +81-45-924-5214

Period of Performance: April/01/2013 – March/31/2015

Abstract:

The purpose of this project is to develop methods to analyze discussion records. We analyze discussion records by two methods – Scene analysis and Logical analysis. Scene analysis focuses on topic change in the record and divides a record into several scenes using Temporal Data Crystallization (TDC). Then, by measuring the polarity of words and the volume of utterance, we estimate the atmosphere and emotion in the scene. Logical analysis focuses on the role of phrases in utterances, and attaches logical labels to phrases based on Toulmin diagram. By these labels, we extract a logical structure of arguments in the discussion, calculate its semantics based on Theory of Computational Argumentation and evaluate various argumentation skills of the discussion record. In addition to discussion analysis, we apply the scene analysis techniques to caption data of TV programs to recognize change of scenes.

Introduction:

In our daily life, we have a lot of opportunities of discussion with others. These discussions are classified into two types – competitive type and cooperative type. In the case of competitive type, the goal of discussion is to defeat other side, and the quality of arguments is important. On the contrary, the goal of cooperative type discussion is to reach the consensus, so not only the quality of arguments but feeling of satisfaction of participants is important.

In the law schools, to educate discussion skills, discussion training is conducted. However, analysis of discussion records is not sufficient because technologies of discussion analysis is still in the low level. For the education of discussion skills, detection of scene change and both logical analysis and emotional analysis of each scene are necessary, and estimating discussion skills from results of these analyses is needed.

However, most traditional researches of discussion analysis focused on only logical aspects. Though it is also important to recognize the turning point in the discussion where topics and atmosphere change, traditional approaches are insufficient.

Therefore, in this project, we aim at developing a method of analysis of discussion record in detail. Our approach uses both scene analysis and logical analysis. For scene analysis, we use Temporal Data Crystallization (TDC), and for logical analysis, we use Speech Act theory and Toulmin Argumentation Model. TDC has been studied in our previous AOARD project, and its performance to recognize key utterance is shown. Toulmin Argumentation Model was proposed by Stephen Toulmin. It represents a logical structure in an argument from view of conclusion, data, warrant and backing. By attaching tags which represent the logical structure to discussion logs, we can extract argument structures (a set of arguments and defeat relation among arguments) and measure the strictness of arguments in the discussion records.

Experiment:

1. Overview of Discussion Record Analysis

The overview of our discussion analysis is shown in Fig. 1. The discussion record is analyzed by two methods – scene analysis and logical analysis. Scene analysis is composed of scene detection which detects points where main topics change (in Section 2.1), and scene feature extraction which estimates atmosphere and emotion by polar analysis and sound volume analysis (in Sections 2.2 and 2.3). In addition to these fundamental methods, to improve the preciseness and to realize real time detection, an advanced scene detection method is introduced in Section 4. Real time scene detection is necessary to analyze TV programs.

Logical analysis is composed of tagging logical roles to sentences in the document (in Section 3.1), calculating its semantics (possible logical consequences of discussion) (in Section 3.3), evaluating discussion skills (in Sections 3.4.1, 3.4.2, 3.4.3 and 3.4.4), and detection of stale mates (in Section 3.5). And an advanced method of calculating semantics which integrates more than one documents is introduced in Section 5.

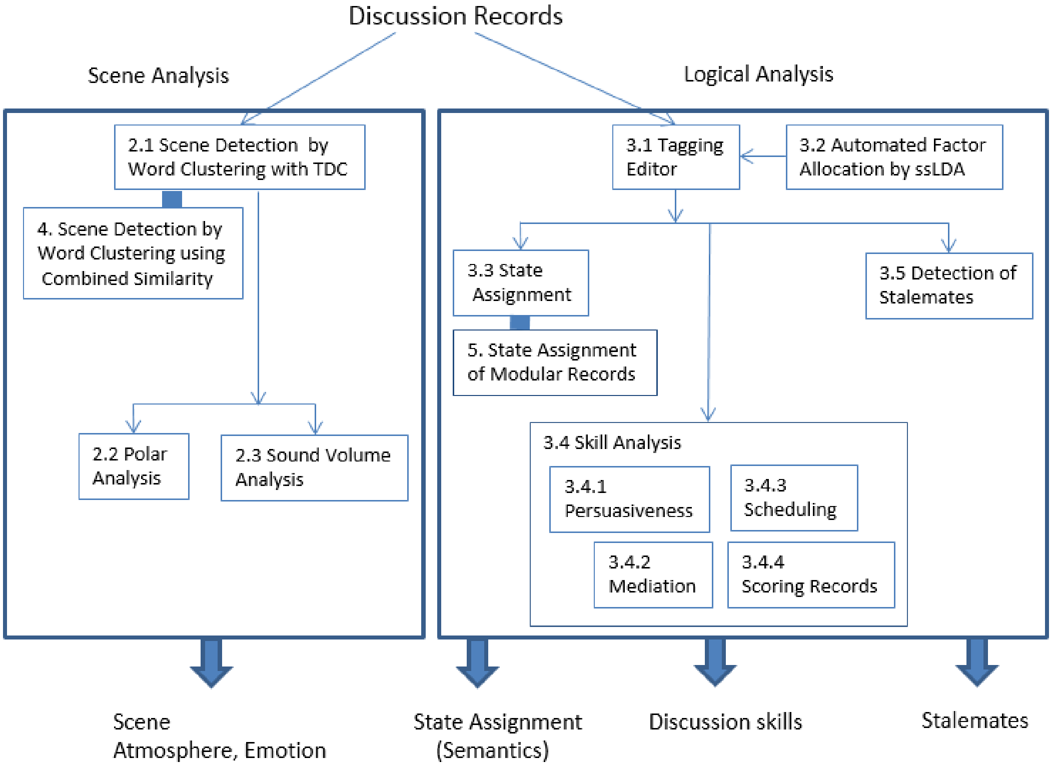

Discussion Records
Scene Analysis
Logical Analysis
2.1 Scene Detection by Word Clustering with TDC
4. Scene Detection by Word Clustering using Combined Similarity
2.2 Polar Analysis
2.3 Sound Volume Analysis
3.1 Tagging Editor
3.2 Automated Factor Allocation by ssLDA
3.3 State Assignment
5. State Assignment of Modular Records
3.5 Detection of Stalemates
3.4 Skill Analysis
3.4.1 Persuasiveness
3.4.3 Scheduling
3.4.2 Mediation
3.4.4 Scoring Records
Scene Atmosphere, Emotion
State Assignment (Semantics)
Discussion skills
Stalemates

Fig. 1 Overview of Discussion Record Analysis

2. Scene Analysis

2.1 Scene Detection by Word Clustering with TDC

To read a huge discussion record and recognize arguments in them takes long time. Before reading the record in detail, if we notified some utterances where topic change occur, such information is expected to reduce the time to read. We detect key utterances by a co-occurrence analysis tool.

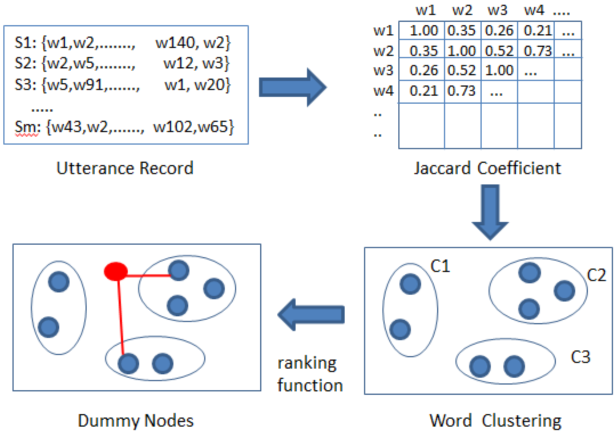
This analysis is based on the idea of Word Clustering with Temporal Data Crystallization (TDC). Word Clustering with Temporal Data Crystallization is performed as follows. Where the discussion record is considered to be a set of S_1, S_2..., and each utterance S_i is considered to be a set of words that appeared, $\{w_1, w_2..., w_n\}$, the method proposed by Maeno et al. defines the distance $d(w_i, w_j)$ between each word as the reciprocal of the Jaccard coefficient. Next, all words that appeared in utterances are clustered into a given number |*C*| (*C*$_0$, *C*$_1$,...,*C*$_{|C|-1}$), by utilizing the K-medoids method. When each word is expressed with a node and words having a high Jaccard coefficient are connected with links, a graph that consists of *n* islands (clusters) can be obtained (Fig.2). Each cluster is probably considered to be a single topic.

Next, for each utterance S_i (i=1,2,…), following ranking functions $I_{av}(S_i)$ is calculated. Here, $c(S_i)$ is the number of words belonging to S_i.

$$I_{av}(S_i) = \sum_{j=0}^{|c|-1}\left(\frac{c(S_i \cap c_j)}{c(S_i)} - \frac{c(S_{i-1} \cap c_j)}{c(S_{i-1})}\right)^2 \qquad (1)$$

Formula (1) is used to find an utterance Si where the rate of clusters changes from the previous utterance S_{i-1}. We select some utterances whose ranking values are relatively high, and for each selected utterance S_k, we insert a dummy node d_k into the graph. The appearance of these dummy nodes suggests that the utterance that corresponds to these nodes refers to several topics. This indicates that other topics are mentioned during the utterance about a certain topic, or a topic is guided to transition to another topic.

The use of dummy nodes provides the possibility of discovering the characteristics that are not expressed on the surface of the utterance record. For example, Maeno has shown the possibility of extracting the hidden intentions contained in an utterance by utilizing dummy nodes. This is because topics that attract attention and interest can be predicted by making utterances that contain related words even without making clear utterances. Fig.3 shows an example of a word clustering graph with dummy nodes.



S1: {w1,w2,......., w140, w2}
S2: {w2,w5,......., w12, w3}
S3: {w5,w91,......, w1, w20}
.....
Sm: {w43,w2,......, w102,w65}

	w1	w2	w3	w4	....
w1	1.00	0.35	0.26	0.21	...
w2	0.35	1.00	0.52	0.73	...
w3	0.26	0.52	1.00	...	
w4	0.21	0.73	...		
..					
..					

Fig. 2 Temporal Data Crystallization

When the discussion record is huge and contains a lot of topics in it, the performance of original word clustering is not so good because a word may belong to more than one cluster. In such cases, a discussion record is divided into several periods (scenes), and for each period, the word clustering is conducted. Consequently, the discussion records are divided into small periods (scenes) hierarchically (Fig. 4). This method is called *Temporal Word Clustering* (TWC).

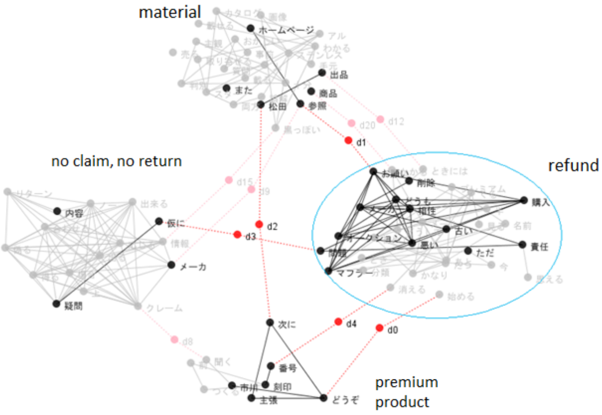


Fig. 3 Word Clustering

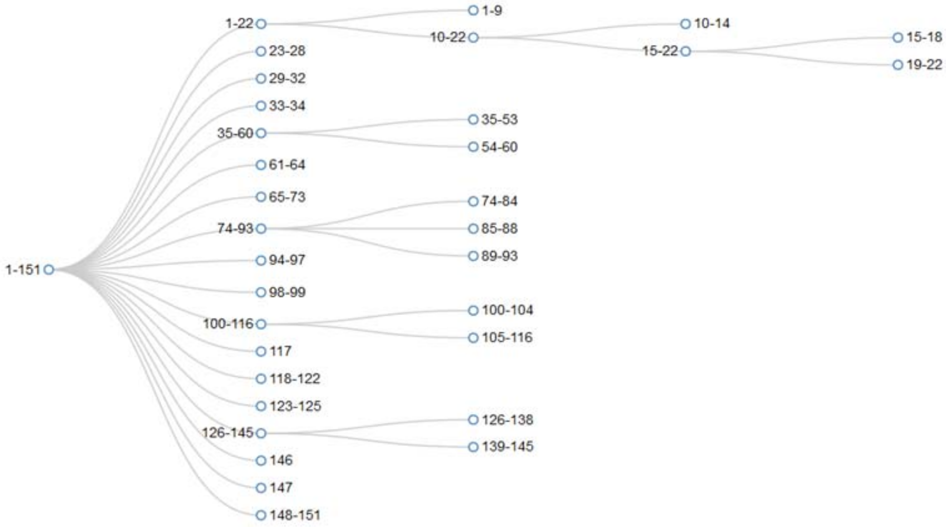


Fig.4 Scene Detection by TWC

2.2 Feature Extraction of Scene by Polarity of Words

After we divide a discussion record into scenes, we measure several features for each scene. Polar Analysis measures atmosphere of the target scene. Takamura analyzed huge documents and attached PN value (1.0 positive ~ 0.0 neutral ~ -1.0 negative) to all words. For example, “agree” and “cheap” are positive words whose PN values are 0.99 and 0.95 respectively. And, “tax” and “difficult” are negative words whose PN values are -0.60 and -0.99 respectively. Feature of polarity of a scene is defined as an average PN value) of polarity of all words which appeared in the scene.

Fig.5 is an example of Polarity analysis of a TV debate program. The upper graph shows the PN value and the lower graph shows heat up scenes which were detected manually.

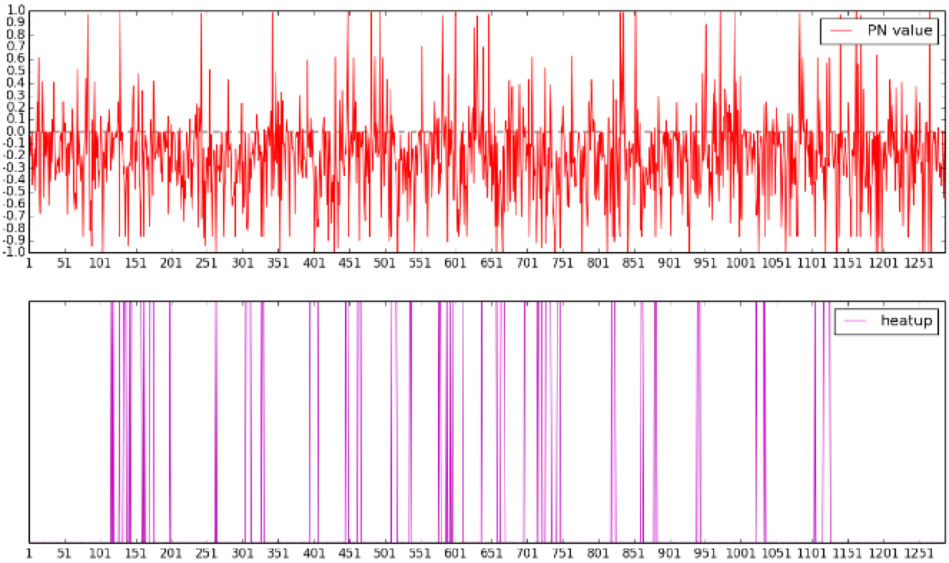


Fig.5　Polarity Analysis

The subject of this debate program is concerning Okinawa's military base. Using TDC method, this TV program is divided into 13 scenes.

ID	1~	44	Headline
	45~	66	Constructing runways
	67~	436	Transferring Okinawa military base
	437~	636	Prime Minister's promise
	637~	694	Deterrent of Okinawa
	695~	723	US-Japan Consulting Meeting
	724~	750	Agreement between US and japan
	751~	879	GDP
	880~	974	Gap between demand and supply
	975~	997	Consumption Tax
	998~	1034	Arguments between Mr.Otsuka and Mr.Takahashi
	1035~	1246	Election
	1247~	1288	Transferring Okinawa military base to Guam

For each scene, we attached labels which represent the level of heat-up. 'A' is very heat-up scene where discussants speak excitingly. 'B' is a medium scene where discussants sometimes speak loudly. 'C' is a quiet scene where discussants speak gently. Fig.6 shows the PN values of each scene. Positive value and Negative values are average of positive values and negative values in the scene.

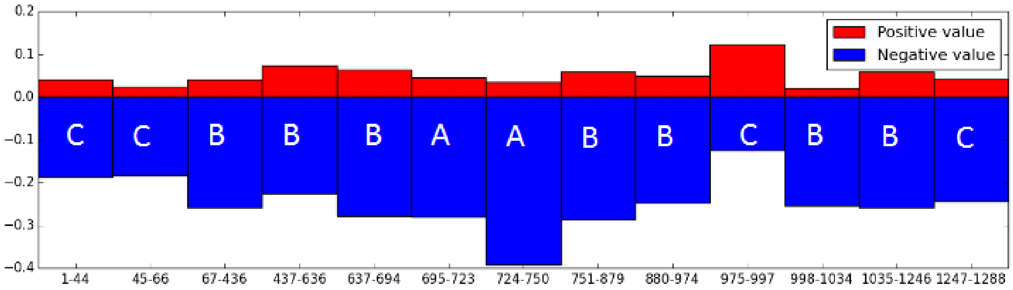


Fig. 6　PN values and Heat-up scene

2.3 Feature Extraction of Scene by Voice

In the previous section, we showed the result of polarity analysis, which uses text data in the discussion record. If the discussion record is a form of movie data, we may extract heat-up scene by volume level of sound. We focused on the fundamental frequency of voice of discussion and classified each scene into heat-up one or not using SVM. The result shows that the precision value, the recall value and the F-value of this method are 0.58, 0.57 and 0.58, respectively.

3. Logical Analysis

The purpose of logical analysis is to give information about argumentation skills evaluation from the discussion record.
At first, we read the discussion record and attach several tags to it by referring to a factor list. The output of this step is a tagged record. Here, factors are axioms, which appear in the discussion. Then, by using these tags, we calculate the semantics of the discussion, measure various features of discussion by comparing other record and recognize several patterns of stalemate, which are used to evaluate argumentation skills.
We will introduce some important concepts.

3.1 Tagging Editor

We developed an editor for constructing diagrams and annotating each utterance in a discussion record with some tags. This tool also visualizes logical structure of the argumentation with diagrams.
Input of the editor is a discussion record and a factor list.

A *factor list* is a set of axioms, which describe facts, events, claims and issue points in utterances. Followings are examples of factors, which may appear in the discussion about the abandonment of nuclear power plants.

F1: Nuclear power is necessary.
F2: Nuclear power is not necessary.
F3: We need a sable energy source.
F4: We can substitute the nuclear power by solar energy generation

Among factors, there exist two types of relationships as follows.

F1 <u>attacks</u> F2.	F2 <u>attacks</u> F1.
F3 <u>supports</u> F1.	F4 <u>supports</u> F2.

We use three
In our system, we attach following three types of tags to each utterance in the discussion record.

(i) An utterance ID, and a speaker ID
(ii) Speech acts
(iii) Argument structure

Speech act denotes the role of the utterance such as *claim*, *argument*, *agreement*, *denial*, *complement*, *close-ended-question(CQ)*, *open-ended-question(OQ)*, *answer*, *demand*, *proposal* and other. CQ is a question whose answer is YES or NO, and OQ is a question which requires some explanation. For example, “Were you pleased to hear that?” is a CQ, and “How did you feel to hear that?” and “Why were you pleased to hear that?” are OQs. In the case of mediation, the mediator is expected to use more OQs than CQs.

When the speech act of an utterance is an argument, furthermore, we recognize the *conclusion* part, the *data* part and the *warrant* part of the Toulmin model, and attach tags for each part. For example, consider the following arguments between Mr. A and Mr. B.

A20> “We need nuclear power plants (F1) because we need huge (F3) amount of energy according to the document X.”
B21> “We don’t need nuclear power plants (F2) because we can replace it by solar power supply (F4) according to the document Y.”

These arguments are represented as a diagram (Fig.7).

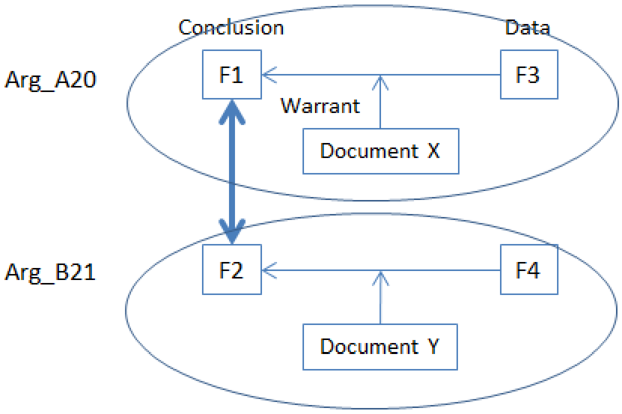
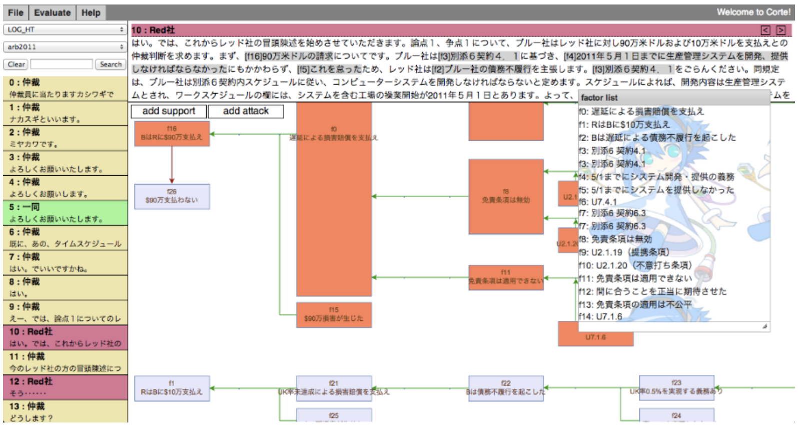


Fig. 7　Visualization of Argument Structure

Diagram data does not initially exist; however if there already exists diagram data with factor list concerning that discussion, users can reuse the data for annotation. Output is annotated log, diagram data and some evaluation report generated by further analyzing tools integrated with this editor. These I/O files are formatted in XML therefore other systems would be able to utilize them.

Screen of this editor (Fig. 8) is mainly consisted of three parts: utterance list (left column), utterance detail (right column top) and diagram editor (right column bottom). When a user clicks an utterance from the list, its total text is displayed on the utterance detail part. On diagram editor part, users edit diagram visually. A box on the screen represents an element. Users add boxes and connect each box to construct diagram. After editing boxes, users annotate part of an utterance with a box, or an element, by dragging mouse on the part of the text on utterance detail and selecting corresponding element on the diagram editor.



0：仲裁
仲裁員に当たりますカシワギで

1：仲裁
ナカスギといいます。

2：仲裁
ミヤカワです。

3：仲裁
よろしくお願いいたします。

4：仲裁
よろしくお願いします。

5：一同
よろしくお願いいたします。

6：仲裁
既に、あの、タイムスケジュール

7：仲裁
はい。でいいですかね。

8：仲裁
はい。

9：仲裁
えー、では、論点1についてのレ

10：Red社
はい。では、これからレッド社の

11：仲裁
今のレッド社の方の冒頭陳述につ

12：Red社
そう……

13：仲裁
どうします？

10：Red社

はい。では、これからレッド社の冒頭陳述を始めさせていただきます。論点1、争点1について、ブルー社はレッド社に対し90万米ドルおよび10万米ドルを支払えとの仲裁判断を求めます。まず、[f16]90万米ドルの請求についてです。ブルー社は[f3]別添6契約4．1に基づき、[f4]2011年5月1日までに生産管理システムを開発、提供しなければならなかったにもかかわらず、[f5]これを怠ったため、レッド社は[f2]ブルー社の債務不履行を主張します。[f3]別添6契約4．1をごらんください。同規定は、ブルー社は別添6契約内スケジュールに従い、コンピューターシステムを開発しなければならないと定めます。スケジュールによれば、開発内容は生産管理システムとされ、ワークスケジュールの欄には、システムを含む工場の操業開始が2011年5月1日とあります。よって、 ムを

f0: 遅延による損害賠償を支払え
f1: RはBに$10万支払え
f2: Bは遅延による債務不履行を起こした
f3: 別添6 契約4.1
f3: 別添6 契約4.1
f4: 5/1までにシステム開発・提供の義務
f5: 5/1までにシステムを提供しなかった
f6: U7.4.1
f7: 別添6 契約6.3
f7: 別添6 契約6.3
f8: 免責条項は無効
f9: U2.1.19（提携条項）
f10: U2.1.20（不意打ち条項）
f11: 免責条項は適用できない
f12: 間に合うことを正当に期待させた
f13: 免責条項の適用は不公平
f14: U7.1.6

Fig.8　Screen Shot of the Tagging Editor

To analyze multiple argumentations using this tool, users work with the following steps:

1. Input a whole discussion record and a factor list.
2. Read an utterance and annotate speech acts of the utterance and select factors which match to the contents of the utterance.　If there is no proper factor in the factor list, the user may register a new factor.
3. If the utterance contains an argument in it, generate a box in the diagram in the screen, and indicate the corresponding part in the utterance, and select the proper factor.
4. Repeat Step2 and Step 3 until finishing annotation of logs to be analyzed.

After finishing annotation, the use is able to check if the tagging is conducted properly by showing the diagram and the sequence of speech act.

3.2 Automated Factor Allocation by ssLDA

To use the tagging editor, the user must read the discussion record and recognize which factor appears in the sentence. To support this step, we developed an automated factor allocation module, which uses the technique of Semi-supervised Latent Dirichlet Allocation (ssLDA).

3.2.1 ssLDA

As an extension of existing supervised topic models, semi-supervised latent Dirichlet allocation makes use of both labeled and unlabeled documents in the corpus, and the generative process, as shown in Fig. 9 is as follows:

1. For each label:
   Draw a multinomial distribution over all words $\beta_{1:K} \sim Dir(\eta)$.
2. For each of M_1 labeled documents with a multi-label set Λ:
   (a) Draw a multinomial distribution over all possible labels $\theta \sim Dir(\alpha)$.
   (b) For each of the N word:
      (i) Sample a label z that is within the multi-label set Λ.
      (ii) Sample a word w from the multinomial probability conditioned on the label z.
3. For each of M_2 unlabeled documents:
   (a) Draw a multinomial distribution over all possible labels $\theta \sim Dir(\alpha)$.
   (b) For each of the N word:
      (i) Sample a label z.
      (ii) Sample a word w from the multinomial probability conditioned on the label z.

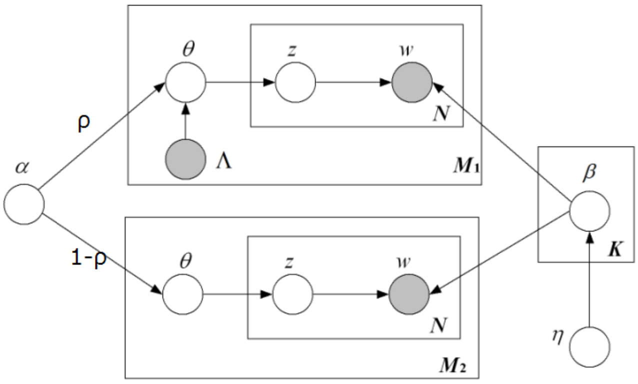


Fig. 9 Graphical illustration of semi-supervised LDA.

The structure shown in Fig. 9 demonstrates that both labeled and unlabeled documents are used to learn a model in ssLDA. Although only w and Λ in labeled documents are treated as observed variables in Figure 10, ssLDA can also incorporate the assignment of labels to certain words in the document, which makes z in Figure 10 representing a label also observable.

3.2.2 Evaluation of ssLDA

In the preceding section, we demonstrated the strategy of transforming the unsupervised LDA into a semi-supervised algorithm. We compare the performance between ssLDA, transductive support vector machines which can be viewed as a special case of semi-supervised learning in discriminative fashion and kNN using two datasets labeled at word level and at document level respectively.

Existing evaluation metrics for multi-label prediction tasks are largely based on two perspectives: (1) document-based, which emphasizes the prediction of each test instance; (2) label-based, where the prediction of each label is focused. In this paper, we investigate F1scores of experiment results under both perspectives. In particular, assume that there are M test documents and K unique labels, the computation for document-based F1 scores break the predictions within each document down into K binary-classification problems, while for label-based F1 scores we first fix a label and then compute the F1 score across all M documents. Finally, we take the average of F1 scores across all documents and labels respectively.

(1) Experimental Settings
We ran experiments on a specific corpus consisting of 116 conversation records extracted from twelve groups of mock mediation each approximately lasting forty minutes. With a vocabulary of 117 words, we count the number of the words appearing in the speech and the corpus is built under a "bag of words" scheme. Despite only 117 words selected as features to represent a speech, there are 17 predefined labels (also called issue point or factors), which are critical in dialogue analysis. Since each label usually corresponds to a certain part of the document and the direct correspondence is very easy to identify, it is natural to build such an assignment between words and labels with ssLDA. On the other hand, SVM-based algorithms in this experiment assign the labels at document level.

(2) Experimental Results
Fig. 10 and Fig. 11 present F1 scores as the number of labeled documents increases obtained by ssLDA, transductive SVM and 1NN under label-based and document-based perspective. In ssLDA, we have two parameters, ρ and a threshold to decide relative labels based on the ranking of all possible labels. The parameter ρ is the ratio of labeled and unlabeled documents (Fig.9), and the threshold is used to select labels. If the weight of each label is less than the threshold, the label is not used. Here we set ρ=0.8, and let the threshold be 0.15. The implementation of a transductive support vector machine we use here is SVM^{light} with fraction of unlabeled examples to be classified into positive class as the ratio of positive and negative examples in the labeled training documents. Besides, the nearest neighbor algorithm is also compared here, and the distance metric is chosen as cosine distance between two documents under the "bag of words" presentation.

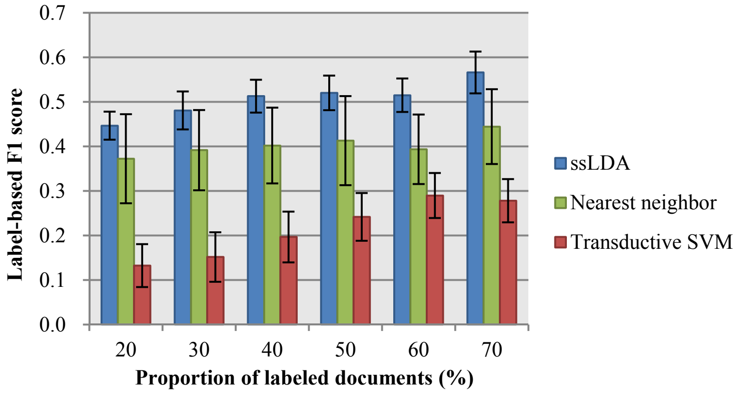


Fig. 10 Label-based F1 scores with respect to different proportions of labeled documents.

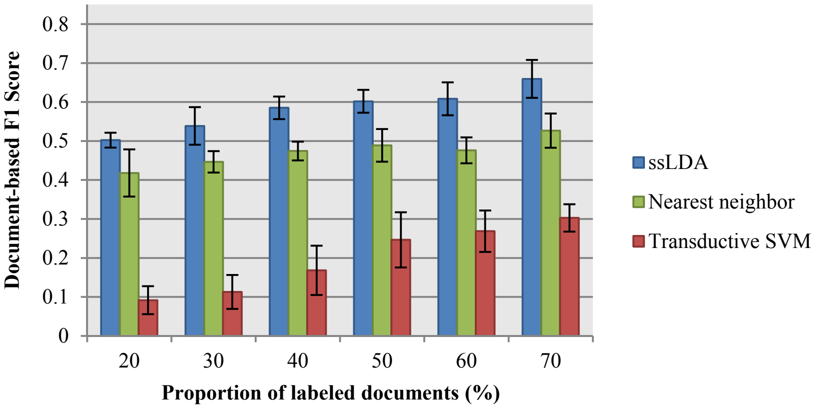


Fig. 11 Document-based F1 scores with respect to different proportions of labeled documents.

From Fig. 10 and Fig. 11 we see that ssLDA achieves a better performance than transductive SVM and the nearest neighbor algorithm in label-based and document-base perspective. Furthermore, it is worth mentioning that SVM-based algorithms seem not working well in this experiment where the size of the vocabulary is fairly small while the total number of labels is relatively large. Although the nearest neighbor algorithm is the simplest one to implement, it still achieves a notably better performance than the transductive SVM.

3.3 State Assignment for Arguments (Semantics of Argumentation)⊆

To define the semantics of arguments, Dung introduced Argumentation Framework (AF). AF is composed of a set of arguments and a set of attack relations between two arguments. AF is considered as a directed graph structure where each node represents an argument and each link represents attack relation. Fig.12 is an example of AF=*(Args, attacks)*, where *Args={A, B, C, D, E}*, and *attacks={(A,B), (B,A), (C,B), (C,D), (D,C), (E,A)}*. In this example, as an argument *E* is not attacked by any argument, *E* always holds. Therefore, an argument *A* doesn't hold because *A* is attacked by *E*. Whether *B*, *C* and *D* hold or not depend on other argument holds or not.

Let consider an argument set *S* ($S \subseteq Args$). If there is no attack relation between members of *S*, *S* is called *conflict-free*. If *S* is conflict-free, and when a member of *S* is attacked by other argument *X* ($X \notin S$), at least one member of another argument set *R* ($R \subseteq Args$) attacks *X*, we say that *S* is *acceptable with respect to R*. For example, in Fig.12, an argument set *S (={B, E})* is acceptable with respect to an argument set *R (={D})* because *B* ($\in S$) is attacked by *C* ($\notin S$) and *D* ($\in R$) attacks *C*.
When *S* is *acceptable with respect to R*, we may think that *R* defends *S*. If *S* is *conflict-free* and *S* is *acceptable with respect to S*, we say that *S* is *admissible*. If *S* is *admissible*, we may think that *S* defends itself.
If an argument set *S* is an admissible set and every acceptable argument with respect to *S* belongs to *S*, *S* is called a *complete extension* of AF. If an argument set *S* is maximal among admissible sets, *S* is called a *preferred extension* of AF. If an argument set *S* is the smallest among complete sets, *S* is called a *ground extension* of AF. A preferred extension and a ground extension are complete extensions.
Let consider following sets of arguments.

S1 = {C, E}, S2 = {B, D, E}, S3 = {E}

S1 and *S2* are preferred extensions of AF, and *S3* is a ground extension.

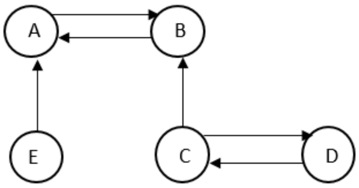


Fig. 12 Example of Argumentation Graph

Based on tagged discussion records, we can easily generate AF, and calculate its semantics explained above. Then, for each argument that appears in the discussion record, we can decide if it is a justified argument, a defensible one or a defeated one. This information is used to measure the strictness of the discussion.

Since tag information contains 'conclusion', 'data' and 'warrant' labels of Toulmin diagram, recognition of arguments is easy. For example, Toulmin diagram of Fig.13 (upper) is considered as following logical formula. Here, *Rs* is a set of strict rules, *Rd* is a set of defeasible rules, and *Ka* is a set of assumption premises.

Rs: { ←b,e }
Rd: { a <= b; b <= c; d<=e }
Ka: { c; e }

Then, arguments and sub arguments are recognized as Fig.12 (lower). This figure corresponds to the following AF (= (Args, attacks)).

Args = { A1: a<=A2; A2: b<=A3; A3: c; A4: d<=A5; A5: e }
attacks= { (A2,A4), (A2,A5), (A5,A2), (A5,A1) }

For the resultant AF, our tool calculates argument semantics such as grounded, stable, complete and preferred extensions. If an argument is included in the grounded extension, or if an argument is included in all preferred extensions, the argument is a justified one.

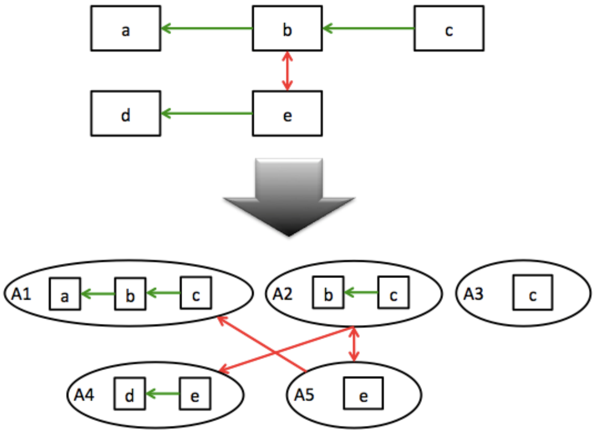


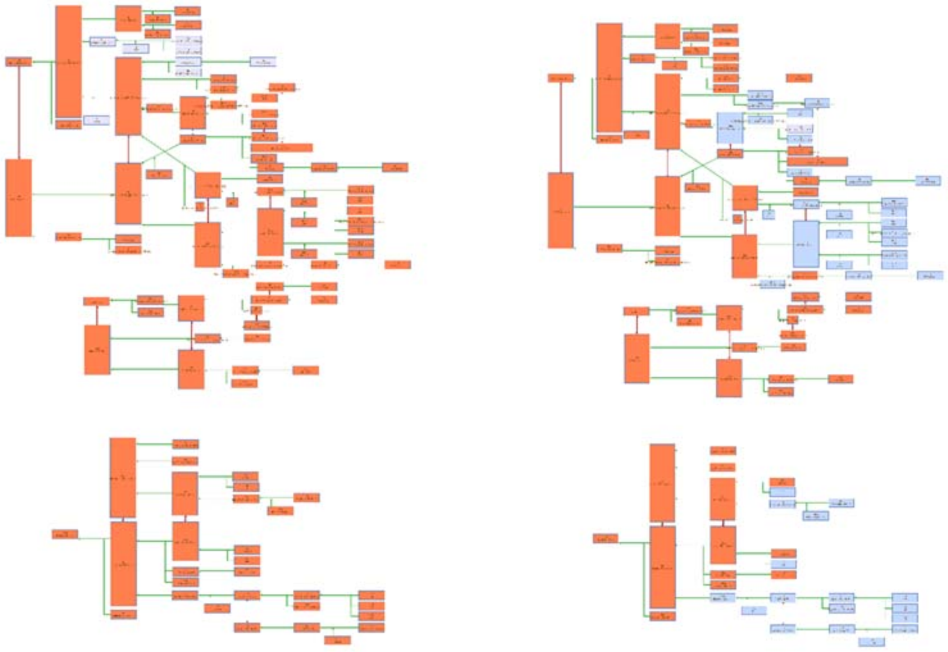
Fig. 13 Conversion from Tags to AF

3.4 Skill Analysis

Multiple discussions are comparable with respect to the shared diagram if those discussions share the same issue. In this section, we describe the method of these comparisons supported by the tool. To compare detailed structure of these argumentations, this tool support to compare the logical structure of discussions, type of mediators and scheduling skill of mediators. And this tool also has a functionality of estimation of evaluation from evaluated logs by calculating similarity of discussions.

3.4.1 Comparison of Logical Structure

By comparing Toulmin diagrams which are obtained from discussion records for the common case, we may evaluate which discussion is discussed in more detail.
We will show an example of arbitration records of Intercollegiate Negotiation Competition in Japan. Diagrams in matchups A and B is shown in Fig. 14, where the same diagrams are shown and referred factors in each matchup are displayed as a colored box in each matchup while pale boxes are not referred. Though this is just a part of that arbitration, strategies or tendencies of the teams are clearly visualized. The upper portion (A) of Fig. 14 shows that these teams tend to give attention to detail of the argument, while that of lower portion (B) omitted them. By these diagram comparisons, users of this tool will be able to measure argumentation skills visually.
As criteria to measure the difference of diagrams, we used indices such as number of support relation, number of attack relations, factor coverage rate and warrant coverage rate. The number of support relations and the number of attack relations are criteria of activeness of discussion. Factor coverage rate measures the elaboration of discussion. Warrant coverage rate measures if each argument is supported by some law. Each coverage rate is calculated by the number of the specified elements per that of total elements. Value of those indices in the three matchups (A, B, C) is shown on Table 1.
And the order of the values in each matchup is $A > B \geq C$. These indices show that teams of matchup A performed more logical discussion than that of matchup B and C. And matchup B is slightly more logical than that of C.
The result of the competition corresponds to this evaluation. A team of matchup A (team X) won a much better prize in the competition. There is a team (we call it team Y) participated in both matchups A and B. Team Y also won a prize (although X got better ranking than Y). In addition, a team (team Z) participated in both matchups B and C. However team Z did not win a prize. Therefore these indices are useful to evaluate discussion skills with respect to logical structure of argumentations.



(A) Referred Elements in Matchup A (B) Referred Elements in Matchup B

Fig. 14 Difference of Referred Factors

Table 1. Strictness of Discussion

Index	A	B	C
Number of support relations	55	39	31
Number of attack relations	28	18	18
Factor coverage rate [%]	93	59	52
Warrant coverage rate [%]	79	36	36

3.4.2 Type of Mediator

Mediators are classified as facilitative type or evaluative type. The facilitative mediator structures a process to assist the parties in reaching a mutually agreeable resolution by asking questions; by validating and normalizing parties' points of view; by searching for interests underneath the positions taken by parties; and by assisting the parties in finding and analyzing options for resolution. On the other hand, the evaluative mediator makes recommendations to the parties, give his or her own advice or opinion as to the outcome of the case, or predict what a court would do in the case. In short, the facilitative mediator is interested in the discussion process, and the evaluative mediator is interested in the conclusion of the discussion. Facilitative mediation is preferred in general.

This tool supports to measure mediators on that point of view. However it is difficult to handle process of building consensus on diagram. Therefore we utilized speaker and speech act tags to measure it.

To measure the mediation type, we propose an index, which is the rate of mediator proposals (number of proposing utterances by the mediator / that of total utterances). The rate and number of utterances in the eight mediation logs in the example case (muffler case) are on Table 2. For comparison, we evaluated degree of facilitative in each log manually and showed it at the bottom of Table 2. In this column, *F* means a facilitative mediator and *E* means an evaluative mediator.

Table 2. Rate of Proposal Utterances by Mediators

	1	2	3	4	5	6	7	8
(1) Total proposal utterances	2	7	2	1	5	4	1	7
(2) Proposal utterances by mediator	0	3	1	0	3	2	0	5
(3) Total utterances	76	68	91	25	70	55	67	62
(2) / (3) [%]	0	4.4	1.1	0	4.3	3.6	0	8.0
Evaluation	F	E	F	F	E	F	E	E

On Table 2, facilitative mediations tend to indicate lower value of the rate than that of evaluative ones. This is partly because facilitative mediators tend to avoid presenting a plan for agreement while evaluative mediators actively try to do that based on the perspective or evaluation of mediators. Therefore functionality of calculating these evaluation indices supports to measure type of mediators.

3.4.3 Scheduling Skill of Mediator

In many cases of mediation, mediators hear detailed issues from the interested parties before discussing agreement plan. This procedure is important for efficient mediation because agreement construction is based on those issues or argumentations. If a participant claims a new fact after beginning discussion of agreements, it should not be allowed because it may cause the delay of discussion of agreements. Therefore mediations shall be clearly separated into issue hearing part and agreement building part. We estimate that when the first proposal of agreement is made, mediations go into the agreement building part. And if participants of mediations rake over old ashes which were treated in issue hearing part, we regard scheduling skills of mediators are not good.

On this point of view, we propose an evaluation index, which is the rate of referred elements in the shared diagram after the first proposal of agreement (number of referred elements after the first proposal / number of utterance after the first proposal). This index describes how often rehashing a matter occurs. If rehashing a matter often occurs, then we regard the mediation process has not progressed smoothly. To calculate this, we utilize speech act and element tags.

Table 3. Number of Referred Elements after the First Proposal

	1	2	3	4	5	6	7	8
(1) Number of referred elements after the first proposal	0	15	0	0	0	3	4	0
(2) Number of utterance after the first proposal	32	53	68	8	24	32	28	28
(1) / (2) [%]	0	28	0	0	0	9	14	0
Evaluation	A	C	B	B	A	C	C	B

To evaluate this index, we use mediation records, and evaluated smoothness of discussion in each log manually which is shown at the bottom of Table 3. On Table 3, it is shown that discussions with nonzero value of the index are not smooth. Therefore it is able to find not very skillful mediators concerning with scheduling skill from those tags.

3.4.4 Clustering of discussion records

When there are many discussion records sharing same issue, detailed analysis of each pairs is time-consuming task. Therefore we prepare functionality of estimation of evaluation based on similarity of discussion logs.

Tanaka showed that mediation logs might be categorized by their similarity based on feature vectors that represent discussion logs. Each dimension of the feature vector is a factor and the value of each dimension is number of each factors referred in the discussion. Similarities are calculated by inner products of each pair of vectors.

Similarity of eight mediation discussions of the same case with respect to factor coverage is shown in Table 4. In this table, values more than 0.80 appear at positions of (1,5), (3,5), (3,7), (3,8), (4,7), (5,7) and (5,8). For comparison, we graded each discussion record as follows.

Grade A:	1, 5
Grade B:	3, 4, 8
Grade C:	2, 6, 7

Among 7 positions listed above, 2 positions matched the grades correctly, 4 positions matched within one grade difference, and 1 position failed to match. For present, this result is not satisfactory. We will try another index for classifies discussion records.

Table 4. Similarity of Mediation Records with Respect to Factor Coverage

	1	2	3	4	5	6	7	8
1		.65	.67	.59	.87	.69	.77	.77
2			.59	.56	.78	.76	.68	.71
3				.75	.81	.61	.85	.80
4					.74	.57	.83	.62
5						.64	.83	.81
6							.71	.73
7								.78
8								

3.5 Stalemate Detection

Stalemate is a contextual situation where both side of the discussions repeat the same claims and the discussion does not proceed any longer. Resolution of stalemate is an important skill of discussion. Since stalemate is contextual, tag based pattern matching is available while word frequency based detection is difficult without enormous volume of annotated discussion logs. Therefore we utilize tags such as utterance ids, elements in the diagram, speech-acts and speakers for detection.

We found a few suspicious patterns of the stalemate from annotated discussion log. These patterns are described as following sequence of tags where there are two speakers A and B and speech act tags are denoted by quoted string. Some other utterance may exist between each step of the pattern, but likelihood of stalemate rise as the sequence of tags in a discussion log match the pattern.

Pattern 1 (no sub argument exists in conflicting arguments):

0. An attack relation exists between arguments X and Y
   (Y attacks X).
1. Speaker A states argument X with its claim and data.
2. Speaker B states argument Y with its claim only.
   And speaker B repeats its claim. (No other argument is stated till the repetition.)
3. After above situation, speaker A make a question such as “Why do you think so?” which is represented by a speech act tag of “Open-ended-question.”
4. Speaker B states data (and warrant) of argument Y.

In this pattern, attacked side is not convinced at all because attacking side does not present the premise of the claim. After referring the data, the discussion would proceed by attacking the premise or

exchanging other arguments if the participants do not have other factors or knowledge concerning with the conflicting argument.
Following sequence of utterance is an example of this pattern in negotiation of two companies Red and Blue. They are planning to receive an order of a project as a joint venture. And Blue is selling their product "tester" to Red simultaneously.

Blue 1: We agreed that both companies take 10% off the price of their product so the proposing price is $0.5 million lower than the target price (e1). If you leave this $0.5 million to us, we will take $0.1 million off the price of our tester for you (e2). Do you accept it?
Red 2: The discount rate shall be equal. (e3)
Blue 3: Is it important to keep equality? (e2)
Red 4: We do not compromise it. (e3)
Red and Blue: (repeat these question and answer)
Red 9: Why do you think so?
Blue 10: The rate of profit share is defined by discussion conducted before (e4). We shall respect this (e5) therefore we do not accept your proposal.

The sequence of tags in the above utterances is on Table 5 (where "open-ended-question" and "close-ended-question" are denoted by "OQ" and "CQ" respectively, and an argument consisting of a claim element ei and a data element ej is denoted by "ei ← ej").

Table 5. An Example of Sequence of Tags in Pattern 1

id	speaker	speech act	Logical structure
1	Blue	proposal	e2 ← e1
2	Red	denial	e3 attacks e2
3	Blue	CQ	e2
4	Red	answer	e3
…			
9	Red	OQ	
10	Blue	answer, argument	e3 ← e4, e5

In this case, Red side does not present data and warrant element (e4 and e5) of the argument (e3 ← e4, e5) and Blue side was not persuaded till last utterance. Utterances annotated with "question" and "answer" tags is waste of time therefore this part shall be short. Existence of these parts in a log or the length of this part might be also a measure of discussion skill.

Pattern 2 (no more facts to argue):

0. An attack relation exists between arguments X and Y
   (X attacks Y and vice versa)
1. Speaker A states argument X with its claim and data.
2. Speaker B states argument Y with its claim and data
3. Speaker A states argument X with its claim and data

Speaker A repeats argument X at Step 3. This is partly because speaker A has no more argument to attack Y. In this case, the discussion would not proceed unless they abandon this topic.
The following sequence of utterance is an example of this pattern in a log of the muffler case. And the sequence of tags is on Table 6.

Seller 25: The buyer should check the goods just after its delivery. However the buyer claimed refund two months later. It is too late (e10) so we do not accept refund (e2). What is the reason of this duration?
Mediator 26: Do you have any reason to be delayed?
Buyer 27: I am working with vehicles so I need a muffler as a spare of it and I bought it. Since the muffler is a spare, I omitted to check the goods. Two months later, I need to use it so I checked it (e38) out and I found that this is not made of stainless (e11). Alster one is low quality and it is unconformable to my car. So I want to return it and to be refunded (e12).

Mediator 28: Did you apply the muffler?
Buyer 29: No I did not. I found that muffler is not stainless-made (e11) so I want refund (e12).
Mediator 30: (Seller), do you have any opinion?
Seller 31: Checking of the goods does not take time (e10). And buyer shall be responsible for their check. Therefore I do not accept refund (e2).

Table 6. An Example of Sequence of Tags in Pattern 2

id	speaker	speech act	Logical structure
1	Blue	proposal	e2←e1
2	Red	denial	e3 attacks e2
3	Blue	CQ	e2
4	Red	answer	e3
…			
9	Red	OQ	
10	Blue	answer, argument	e3←e4, e5

In this case, the argument (e2 ← e10) appears at utterance 25 and 31. And a conflicting argument (e12 ← e11) appears between them. This discussion becomes stalemate at utterance 31 because the seller does not present new factors to support its argument or attack opposite argument.

Pattern 3 (too much severe request):

1. Speaker A makes a proposal about an agreement plan.
2. Speaker B denies it. (No "agree" utterance exists between current and the last "propose" utterance)
3. Repeats above steps.

This pattern is found at agreement building part of mediation log. The following sequence of utterance is an example of this situation:

Speaker A: The opposite side demands a muffler made of stainless with the same performance as the alster one. Do you accept it? ("proposal")
Speaker B: The price of stainless muffler is much higher than alster one. So we do not accept it. ("denial")
Speaker A: How about refund? The opposite side accepts writing down the price. Do you accept it? ("proposal")
Speaker B: We accept to replace the muffler with equivalent one, but we do not accept refund. ("denial")

In this situation, answering side does not compromise at all. Therefore mediators shall make change the point of view of parties interested. This tool supports to detect stalemate by matching of the above patterns from annotated logs. However there may be many other patterns found in the situation of stalemate, sequence of tags might be useful to detect them or other specific situations.

4. Advanced Scene Analysis – Scene Detection by Word Clustering Using Combined Similarity

4.1 Overview of Advanced Segmentation

In Section 2.1, we introduced a method of scene detection based on word clustering by TDC, and showed its effectiveness. However, when we apply this method to scene detection of TV programs, two problems exist. First one is that when scene changes occurs frequently and few sentences are included in a scene, the performance of detecting scenes decreases. Second problem is that the real time scene detection is impossible because TDC needs clustering of all words which appeared in the document.

To overcome these problems, we developed an advanced scene detection method. The proposed method comprises following three steps.

[1st Step]: Preprocessing, such as tokenization, stemming, and part of speech (POS) filtering, is performed. By this step, each sentence in the input document is changed into a set of words.

[2nd Step]: Calculation of the combined similarity between two words is performed. Combined similarity consists of the semantic similarity and the collocation similarity. Semantic similarity is obtained from the internet using word2vec. Word2vec computes the distributed representations of words from huge datasets. The semantics and relationships between words are embedded in the vector space through model training. When the model is well trained, it is possible to identify similar words in terms of semantics to measure the cosine similarity between words. Collocation similarity is obtained from the target documents. It means how similar frequency of words appearance around those.

[3rd Step]: We calculate the distances between sentences and perform the clustering of sentences.

4.2 Advanced Segmentation

4.2.1 Preprocessing

A document is tokenized into a word stream by morphological analysis, then the word stream is filtered by POS. We only use nouns, proper nouns, verbs, and adjectives, and the Porter stemming method is applied.

Words in a document are denoted in the form of the word stream (w_1, w_2, … , w_z). In the word stream, the same word may appear more than once.

4.2.2 Word Clustering

In domain-independent text segmentation, a lack of background knowledge between each word obtained from within a single document is a problem that leads to poor clustering results. To address this problem, we use word2vec. Each word in a document is input to the word2vec model, and the cosine similarity between words is measured. The obtained matrix is referred to as the semantic similarity matrix S_{sem}.

$$Ssem = \begin{pmatrix} sim_s(w_1, w_1) & \cdots & sim_s(w_1, w_n) \\ \vdots & \ddots & \vdots \\ sim_s(w_n, w_1) & \cdots & sim_s(w_n, w_n) \end{pmatrix}$$

$$sim_s(w_1,w_1) = sim_s(w_2,w_2) = \ldots = sim_s(w_n,w_n) = 1.0$$

The projected word set is represented by {w_1, w_2, …, w_n} in which all words are unique.

Furthermore, we calculate the collocation similarity matrix S_{col}. We count the frequencies of words appearing near the central word of a window via sliding. For example, if a window size of 2, w_i's initial collocation vector is represented by {w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}}. Available words are counted when the window reaches the edge of the word stream. Therefore, w_1's initial collocation vector is represented by {w_2: 1, w_3: 1}. If the same word appears in the word stream, a collocation vector is added. To measure the cosine similarity between collocation vectors and obtaining collocation similarity matrix S_{col}.

$$Scol = \begin{pmatrix} sim_c(w_1, w_1) & \cdots & sim_c(w_1, w_n) \\ \vdots & \ddots & \vdots \\ sim_c(w_n, w_1) & \cdots & sim_c(w_n, w_n) \end{pmatrix}$$

$$sim_c(w_1,w_1) = sim_c(w_2,w_2) = \ldots = sim_c(w_n,w_n) = 1.0$$

We then combine the semantic and collocation similarity matrices as follows.

$$S_{combined} = \alpha\frac{(1+S_{sem})}{2} + (1-\alpha)S_{col}$$

Here, α is the mixture ratio. S_{sem} is scaled between zero and one. We use matrix $S_{combined}$ as an input for word clustering.

As $S_{combined}$ does not satisfy the triangle inequality, we need to apply the clustering algorithm which allows unsatisfied triangle inequality. Moreover, in text segmentation, the number of topical clusters cannot be predefined. Thus, we employ affinity propagation, which takes similarity measures between pairs of data points as input.

Affinity propagation can handle similarity that is not symmetric or does not satisfy triangle inequality and obtain the number of clusters automatically. Moreover, this does not depend on random initialization, such as k-means. Thus, affinity propagation is suitable for our intended purpose.

4.2.3 Segmentation

(1) Distances between sentences with topical clusters frequencies based on EMD

We obtain topical clusters about a document via word clustering. Then, the proposed method measures similarity between each sentence based on EMD (Earth Mover's Distance) with the frequencies of topical clusters to capture the correlation of those clusters.

The EMD is a metric between two distributions defined as the minimum amount of work required to change one signature into another. The notion of work is based on the user-defined ground distance which is the distance between two features. Computation of EMD is based on a solution to the transportation problem, which can be formalized as follows.

Let $P = \{(p_1, w_{p1}), \ldots, (p_m, w_{pm})\}$ be the first signature with m clusters, where p_i is the cluster representative and w_{pi} is the weight of the cluster. Let $Q = \{(q_1, w_{q1}), \ldots, (q_n, w_{qn})\}$ be the second signature with n clusters.

Matrix D is the ground distance matrix, where d_{ij} is the distance between clusters p_i and q_j. Matrix F is the flow matrix, where f_{ij} is the flow between p_i and q_j that minimizes the overall cost.

$$WORK(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}$$

$$f_{ij} \geq 0 \qquad 1 \leq i \leq m,\ 1 \leq j \leq n$$

$$\sum_{j=1}^{n} f_{ij} \leq w_{p_i} \qquad 1 \leq i \leq m$$

$$\sum_{i=1}^{n} f_{ij} \leq w_{q_j} \qquad 1 \leq j \leq n$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j}\right)$$

Once the transportation problem is solved, and the optimal flow F is determined, the EMD is defined as the resulting work normalized by total flow as follows.

$$EMD(P,Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}$$

We define the distance between sentences on the basis of EMD.

The ground distance matrix D consists of the distance between topical clusters pi and q_j as follows.

$$d_{ij} = \frac{1}{|p_i||q_j|} \sum_{w_1 \in p_i} \sum_{w_2 \in q_j} (1 - sim_{combined}(w_1, w_2))$$

This is commonly called the group average method. If EMD is a low value, a sentence P is topically similar to a sentence Q.

(2) Segmentation with DP based on EMD

Text segmentation can be implemented efficiently with Dynamic Programming (DP) techniques. Before explaining the proposed DP approach, we introduce the base DP method proposed by Fragkow.

(2.1) Fragkou DP (traditional DP)

Suppose a document contains N sentences and has a vocabulary of W distinct words. Consider the N×W matrix A defined as follows:

$$A_{n,w} = \begin{cases} 1 & \text{if the w-th word appears in the n-th sentence,} \\ 0 & \text{else} \end{cases}$$

Fragkou et al. define the A × A similarity matrix *D* between sentences of the document as follows.

$$D_{m,,n} = \begin{cases} 1 & \text{if } \sum_{w=1}^{W} A_{m,w}\, A_{n,w} > 0 \\ 0 & \text{if } \sum_{w=1}^{W} A_{m,w}\, A_{n,w} = 0 \end{cases}$$

$D_{m,,n} = 1$ if the m-th and t-th sentences have at least one common word. Good segmentation should maximize the density of 1's in the submatrices of *D*, which correspond to segments. Their considerations can be formalized by defining the segmentation cost function *J* as follows.

$$J = \sum_{k=1}^{K} (\alpha \cdot G(\frac{n_k - n_{k-1} - \mu}{\sigma}) - (1-\alpha)\frac{\sum_{m=n_{k-1}+1}^{n_k} \sum_{n=n_{k-1}+1}^{n_k} D_{m,n}}{(n_k - n_{k-1})^r})$$

The total segmentation cost *J* is the sum of the costs of the K segments, and the cost of each segment is the sum of two terms. The first term is a length cost function obtained by prior knowledge about the average segment length μ. The second term is the generalized density of a segment.

(2.2) Proposed DP

Here, we extend the composition of similarity matrix *D* between sentences. We compose connectivity costs matrix *C* from EMD, which is predefined rather than using common word existence as follows.

$$C_{P,Q} = \begin{cases} 0 & \text{if } P = Q, \\ 1\text{-}EMD(P,Q) & \text{if } P \neq \emptyset \text{ and } Q \neq \emptyset \\ mean(C_{P-1,Q}, C_{P+1,Q}) & \text{if } P = \emptyset \text{ or } Q = \emptyset \end{cases}$$

Here, *EMD* is distance between sentences *P* and *Q*; therefore *1 - EMD* is the connectivity cost between sentences. A short sentence usually contains no term after preprocessing. If there are no terms in the sentence, calculate the mean around sentence for smoothing. This is based on the assumption that proximal sentences are similar to the topic.

Unlike artificial datasets, it is difficult to obtain prior knowledge from real datasets about segments such as the number of segments and their average size. Therefore, we redefine the following segmentation cost function *J'* as follows.

$$J' = \sum_{k=1}^{K} \frac{\sum_{P=n_{k-1}+1}^{n_k} \sum_{Q=n_{k-1}+1}^{n_k} C_{P,Q}}{(n_k - n_{k-1})^r}$$

This function is obtained by removing the first term of the function *J*, by replacing *S* with *C* and by setting *r = 1*. We maximize this cost through DP techniques.

4.3 Evaluation of Advanced Segmentation

4.3.1 Datasets

We evaluated the proposed method using two datasets. The first is a dataset in which each document is a chapter selected from a medical textbook. This dataset is often used as a benchmark in text segmentation.

Here, the task is to divide each chapter into the sections indicated by the author. This dataset contains 227 chapters and 1136 sections.

For the second dataset, we prepared transcripts from one month of televised news program (NHK News 7, an early evening Japanese news program). The reference boundary indicates a topic change in a program, which was annotated manually. This dataset contains 31 programs and 318 boundaries.

4.3.2 Metrics of Evaluation of Scene Detection

Precision and recall are well known criteria to measure the effects of estimation methods. However, in the case of text segmentation, precision and recall are insufficient because whether the predicted boundary is near or far from a reference boundary must be considered. To overcome this problem, we employ P_k and *WindowDiff*, which are widely used metrics in text segmentation. The value of k is half the average reference segment size.

These metrics compute penalty via sliding window size of k. It is described that *1 - WindowDiff* is the effective accuracy of text segmentation.

4.3.3 Experiment Settings

As we explained, our method requires a trained word2vec model. The word2vec model is trained using all current revisions of articles from Wikipedia as of August 2014. We remove meta tags and unnecessary contents, such as lists, titles, template notations, redirect destinations, symbols, and article category information from the Wikipedia dumps. We then convert the resulting data to plain text, which is then tokenized. Stemming is then applied to the data to obtain the words that will be used to train the model. We use a skip-gram model with the following parameters:

window size,	5;
frequency at which words are cut off,	5;
random downsampling threshold,	10^{-5};
vector dimensionality,	100.

4.3.4 Results of Clinical dataset

(1) Relationship between Mixture Ratio α and Collocation Window Size

To know the effects of the mixture ratio parameter α, we shift α and three collocation windows with varying sizes. When α = 0.0, we consider only collocation similarity. For α = 1.0, we consider only semantic similarity. The results are shown in Figs. 5.

P_k and *WindowDiff* demonstrate better performance at α = 0.65 and a collocation window size of 5. Note that these behaviors are similar regardless of collocation window size. The form of the obtained graph resembles a valley, which indicates that either collocation similarity or semantic similarity is insufficient, and the combined similarity is more suitable to our purpose.

(2) Comparison with other approaches

To demonstrate the effectiveness of the proposed method, we compared it to several previous approaches;

As is shown in Table 7, the proposed method clearly outperforms other domain-independent approaches for both P_k and *WindowDiff*. Therefore, we consider the proposed method a state-of-the-art domain-independent approach. On the other hand, compared to domain-dependent approaches, the proposed method is slightly inferior to TSM and NTSeg with P_k, however, it demonstrates good performance with *WindowDiff*. Therefore, the proposed method is on an even level with domain-dependent approaches. In addition, in the range of α between 0.7 and 1.0, proposed method demonstrates good performance with any collocation window size.

Table 7 Comparison of Performance of Scene Detection

Method	P_k	*WD*	domain-independent
C99	38.7	39.7	Yes
U100	37.0	37.6	Yes
LCseg	37.0	38.5	Yes
BAYESSEG	33.9	35.3	No
Topic Tiling	31.9	34.7	No
TSM	**30.6**	34.5	No
NTSeg	30.9	32.7	No
Our Method	31.3	**31.5**	Yes

4.3.5 Results of News program transcripts

(1) Result of Detection of News Programs

The results of segmentation of 31 News programs are shown in Table 8. Both P_k and *WindowDiff* demonstrated better performance at α = 0.8 and collocation window size of 10, 15.

Table 8 Result of Segmentation of News Program

α	P_k	*WD*	Precision	Recall
0.0	33.4	35.4	50.0	31.7
0.8	10.8	14.6	83.6	65.2
1.0	17.5	20.3	86.9	48.7

Generally, rough clusters generate clear topics, but reduce the boundary detection rate. Thus, there is a trade-off between number of clusters and text segmentation performance. However, the proposed method is compatible with both easily understanding topics and boundary detection rate.

(2) Result of Real Time Detection of News Topic

Fig. 15 shows a snapshot of an NHK News program with caption data. We applied our segmentation method to the caption data.

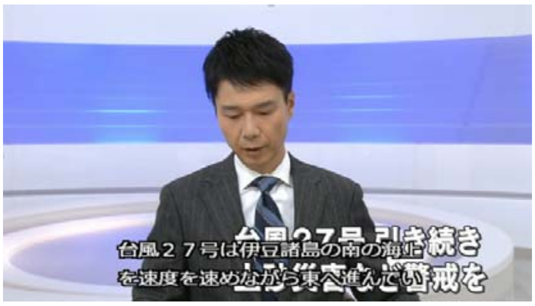


Fig. 15　Caption Data of TV program

This experiment is conducted every 30 seconds incrementally.　Fig. 16 shows the result of real time segmentation for this NHK news program. For example, in Fig. 16, from the beginning of the News to 300 seconds, 36 sentences are included, and during this period, P_k value is 0.033 and *WindowDiff* is 0.167.

Though P_k value of the real time segmentation rises and falls during the program, the value is not so bad even in the worst case. The reason why our advanced method fits for real time segmentation is that it uses combined similarity. The parameter of combined ratio α is usually high (0.8~0.9), so the performance of clustering is not affected by the amount of input document so much.

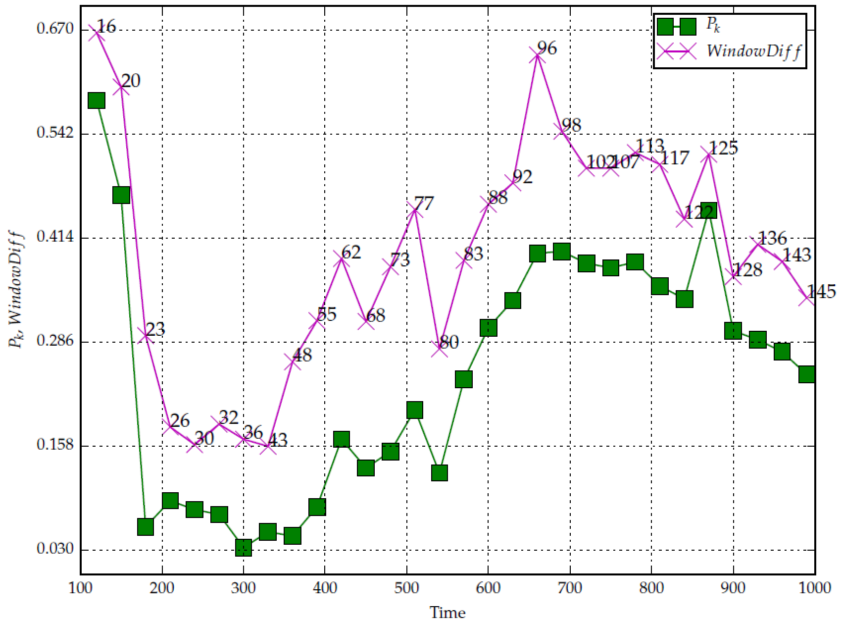


Fig. 16 Result of Real Time Detection of News Program (α=0.8)

5. Advanced Logical Analysis

5.1 Introducing Module Structure

In Section 3, we introduced logical analysis based on Argumentation Framework (AF). However, when the discussion subject is complex, there are several topics and for these topics, the discussions are conducted separately. In such case, integrating these records into one and constructing one AF needs much cost. Instead, if we extract an AF for each record independently, and calculate the semantics of total AF from each semantics, we can reduce the calculation cost.

We show why we employ a module structure in AF using a simple example case. Fig. 17 shows the AF structure of the discussion about the topic of restart of nuclear reactors in Japan. There are 9 arguments {*A, B, C, D, E, F, G, H, I*} and 11 attacks { *(A,B), (B,A), (C,A), (F,C), (F,G), (G,F), (D,B), (D,E), (E,D), (H,E), (I,E)* }.

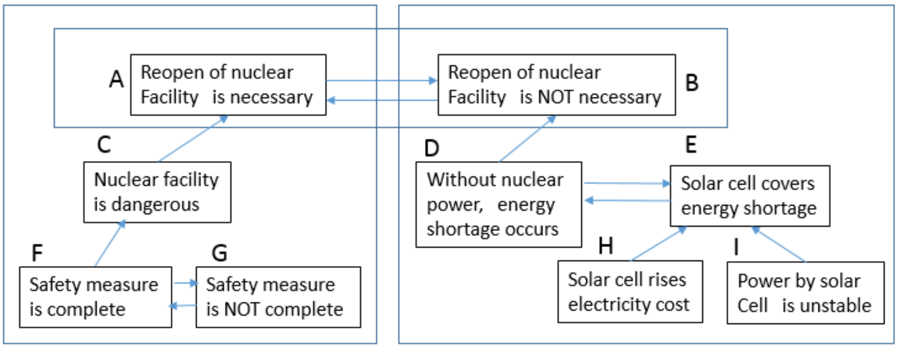

A
Reopen of nuclear Facility is necessary
Reopen of nuclear Facility is NOT necessary
B
C
Nuclear facility is dangerous
D
Without nuclear power, energy shortage occurs
E
Solar cell covers energy shortage
F
Safety measure is complete
G
Safety measure is NOT complete
H
Solar cell rises electricity cost
I
Power by solar Cell is unstable

Fig. 17 An example of AF represents real complex debates

This example contains three kinds of discussion – industry policy issue *{A, B}*, security issue *{C, F, G}* and economy issue *{D, E, H, I}*. Usually, the industry policy issue is discussed by statesmen, the security issue is discussed by scientists and the economy issue is discussed by economists separately. Therefore, there may be three discussion records.

The conclusion of discussion of the security issue *{C, F, G}* affects the reliability of *A*, and the conclusion of discussion of the economy issue *{D, E, H, I}* affects the reliability of *B*. Furthermore, this example shows the conclusion of discussion of the security issue and that of the economy issue don't affect each other.

When the graph is huge, it is useful to divide a huge graph into small graphs for each issue. However, the original AF theory doesn't have a function of integrating local AFs. Therefore, we employed a module structure to AF theory, and constructed *Module Based Argumentation Framework* (MBAF).
In MBAF, we think AF structure in Figure 16 consists of three modules – M_1, M_2 and M_3. Followings are arguments and attack relations of each module.

M_1: Args = {*A, B*}
attacks = {*(A, B), (B, A)*}
M_2: Args = {*A, C, F, G*}
attacks = {*(C, A), (F, C), (F, G), (G, F)*}
M_3: Args = {*B, D, E, H, I*}
attacks = {*(D, B), (D, E), (E, D), (H, E), (I, E)*}

An argument *A* is contained in two modules, M_1 and M_2, and an argument *B* is contained in two modules in M_1 and M_3. We think M_1 is higher than M_2 and M_3 because M_1 discusses the main issue and M_2 and M_3 discuss sub issues which decide the reliability of arguments of main issue. Therefore, the semantics of total AF may be obtained by calculating semantics of M_2 and M_3 at first and then by calculating that of M_1.

5.2 Definition of MAF

In each module, an AF is constructed. Instead of the original AF, we call an AF in a module as *Modular Argumentation Framework* (MAF), because the semantics of an AF in a module *M* is affected by that of an AF of lower modules.

Definition *Modular Argumentation Framework* (MAF)
MAF is a 4-tuple *MAF* = (*Args, attacks, Status, ST*). *Args* is a set of arguments, and *attacks* is a set of attack relation which satisfiy $attacks \subseteq Args \times Args$. *Status* is a set of statuses of reliabilities. In this paper, we define *Status* = {*sk, cr, unc, def*} where *sk* means skeptical reliability, and *cr* means credulous reliability, *unc* means uncertain, and *def* means defeated. *ST* is the function which allocate Status to each argument: *Args→Status*.

If a *MAF=(Args, attcks, Status,ST)* is in the lowest module, for any argument $A \in Args$, *ST(A)=sk*. If a *MAF* is not in the lowest module, *ST(A)* is defined based on MAF in the direct lower module, *MAF'=(Args',attcks',Status,ST')*.

(1) *ST(A) = sk* iff either of the following holds for every direct lower module *MAF'*.
(i) *A* is a member of *MAF grounded extension* of *MAF'*.
(ii) *A* is not a member of *Args'* .

(2) *ST(A) = def* iff the both of the following holds for at least one direct lower module *MAF'*.
(i) *A* isn't a member of *MAF complete extension* of *MAF'*.
(ii) There is at least one *MAF complete labelling* of *MAF'* in which A is labelled *out*.

(3) *ST(A) = unc* iff the both of the following holds for at least one direct lower module *MAF'*.
(i) *A* isn't a member of *MAF complete extension of MAF'*.
(ii) *A* is labelled *undec* in every *MAF complete labelling* of *MAF'*.

(4) *ST(A) = cr* iff all of the following holds for every direct lower module *MAF'*.
(i) *A* is a member of *MAF complete extension* of *MAF'*.
(ii) *A* isn't a member of *MAF grounded extension* of *MAF'*.

MAF complete extension and *MAF complete labeling* is explained in the next section.

5.3 Semantics of MAF

Let *MAF* be (*Args*, *attacks*, *Status*, *ST*). The extensions in the MAF semantics are defined using the notion of Caminada labelling.

Definition *MAF Labelling*
A MAF labelling, *L*, is a total function.
L: $Args \rightarrow \{in, out, undec\}$,
And three sets of arguments are defined as follows.
$in(L) = \{A \mid L(A)= in\}$, $out(L) = \{A \mid L(A)= out\}$, $undec(L) = \{A \mid L(A)= undec\}$.

MAF labelling is another representation to define an argument set. For example, *S1, S2*, and *S3* in Fig. 13 in Section 3.3 is represented as follows.
$S1= \{ L(A)=out, L(B)=out, L(C)=in, L(D)=out, L(E)=in \}$
$S2= \{ L(A)=out, L(B)=in, L(C)=out, L(D)=in, L(E)=in \}$
$S3= \{ L(A)=out, L(B)=out, L(C)=out, L(D)=out, L(E)=in \}$

Definition *MAF Complete Labelling*
L is a *MAF complete labelling* iff for each argument $A \in Args$, followings hold:
(i) If $ST(A) = sk$ and *A* isn't attacked by any argument in the direct lower module, then *A* must be labelled *in*.
(ii) If $ST(A) = cr$, then *A* must be labelled *in* or *out* or *undec*.
(iii) If $ST(A) = unc$, then *A* must be labelled *undec*.
(iv) If $ST(A) = def$, then *A* must be labelled *out*.
Furthermore, each argument which doesn't correspond to any of the condition (i), (ii), (iii) and (iv) follows any one of the followings.
(v) If argument *A* is labelled *in,* then all its attackers are labelled *out*.
(vi) If argument *A* is labelled *out,* then it has at least one attacker that is labelled *in*.
(vii) If argument *A* is labelled *undec,* then it has at least one attacker that is labelled *undec* and it does not have an attacker that is labelled *in*.

Definition *MAF complete extension*
If *L* is a *MAF complete Labelling*, a set of argument, *in(L),* is a *MAF complete extension*.

Definition *MAF ground extension*
A *MAF ground extension* is the smallest set among *MAF complete extensions*.

Difference between the AF semantics and the MAF semantics is that the semantics of AF in some module is affected by the AF semantics in the direct lower module. By employing MAF semantics, we can calculate the AF semantics of the highest level module using the AF semantics in lower modules.

Let consider two modules M_1 and M_2, where M_1 is higher than M_2. And let $AF_1=(Args_1,attacks_1)$ and $AF_2=(Args_2,attacks_2)$ be AFs in M_1 and *M2*. For the integrated $AF_0=(Args_1 \cup Args_2, attacks_1 \cup attacks_2)$, following theorem holds.

Theorem
If an argument A is a member of some *complete extension* in AF_0, then A is a member of *MAF complete extension* in AF_1.

Results and Discussion:

We developed two methods of analyzing discussion records.

(1) Scene analysis method consists of scene detection (segmentation) and feature extraction for each scene. By measuring features of each scene, we can estimate atmosphere and emotion of each scene. In the first year of this project, we applied this method to debate TV programs where a few people join the debate concerning several topics, and showed we can estimate heat-up scenes in some precision level. However, the precision level is not satisfactory.
In the second year, we developed an advanced segmentation method. This method uses combined similarity, which uses not only co-occurrence features in the document but similarity obtained from internet. By this method, we showed precision of segmentation improved. And we applied this method to TV News programs, and we tried the real time scene analysis, which analyzes the TV programs in the real time using the closed captions.

(2) Logical analysis method extracts logical structure of discussion, and measures semantics and discussion skills. In the first year, we developed a tagging editor by which we attach logical tags to phrases in the discussion records. We showed these tag information is useful to evaluate the semantics and discussion skills. We have studied this method with professors of law schools. They wish our system becomes a practical discussion analysis tool.
In the second year, we extend the basic theory of computational argumentation. To cope with a huge and complex discussion record, we introduced a module structure into the Argumentation Framework theory, and proposed extended AF theory.

List of Publications and Significant Collaborations that resulted from your AOARD supported project: In standard format showing authors, title, journal, issue, pages, and date, for each category list the following:
a) papers published in peer-reviewed journals,
b) papers published in peer-reviewed conference proceedings,

[1] Shumpei Kubosawa, Kei Nishina, Masaki Sugimoto, Shogo Okada, and Katsumi Nitta, A Discussion Training System and Its Evaluation, Proc. International Conference on Artificial Intelligence and Law, pp. 197-201, 2013.
[2] Youwei LU, Shogo OKADA and Katsumi NITTA, Semi-supervised Latent Dirichlet Allocation for Multi-label Text Classification, IEA/AIE 2013: pp. 351-360, 2013.
(This paper is also included in "Recent Trends in Applied Artificial Intelligence", Lecture Notes in Computer Science Volume 7906, 2013, pp. 351-360)
[3] Kei Nishina, Shogo Okada and Katsumi Nitta, Module Based Argumentation Framework for Analysis of Actual Discussion records. Proc. International Workshop of Juris Informatics (Jurisin 2014), 2014.
[4] Makoto Sakahara, Shogo Okada and Katsumi Nitta, Domain-independent Unsupervised Text Segmentation for Data Management, International Workshop on Designing the Market of Data (MoDat 2014), 2014.

c) papers published in non-peer-reviewed journals and conference proceedings,

[5] Makoto Sakahara, Shogo Okada, Katsumi Nitta, Extraction of Emotional Utterances and Discussion Analysis Based on Multi Modal Information, Annual Conference of JSAI, 2013 (in Japanese)

d) conference presentations without papers,
e) manuscripts submitted but not yet published, and
f) provide a list any interactions with industry or with Air Force Research Laboratory scientists or significant collaborations that resulted from this work.

Attachments: Publications a), b) and c) listed above if possible.

DD882: As a separate document, please complete and sign the inventions disclosure form.

Important Note: If the work has been adequately described in refereed publications, submit an abstract as described above and refer the reader to your above List of Publications for details. If a full report needs to be written, then submission of a final report that is very similar to a full length journal article will be sufficient in most cases. This document may be as long or as short as needed to give a fair account of the work performed during the period of performance. There will be variations depending on the scope of the work. As such, there is no length or formatting constraints for the final report. Keep in mind the amount of funding you received relative to the amount of effort you put into the report. For example, do not submit a $300k report for $50k worth of funding; likewise, do not submit a $50k report for $300k worth of funding. Include as many charts and figures as required to explain the work.